File(ফাইল)

Some Important Basic.

1.Introduction to file.

2. Writing a file using fputc().

3. Writing a file using fputs()

4. Writing to file using fprintf()

5. Reading a file using fgetc()

6. Reading a file using fgets()

7. Reading a file using fscanf()

8. Students details by using file.

9. Preprocessor #define

10. Preprocessor creating header file

11(1).writing to a file using fprintf()

11(2). Writing to a file using fprintf()

12. Reading from a file using fscanf()

13. নিজে নিজে হেডার ফাইল তৈরি করা

14. Sum of two number using file

15. Copy a image from input.img to output.img

16. Fingign sum and aveage using file-1

17. Finging sum and average using file-2

18. Usng of fseek and perror function.

19. Finding a file size using ftell()

20. Remove a file

21. Payroll

22. Read character from a file

23. Numbering the lines

24. A voting problem

25. A Geography Quiz Problem

26. Word frequency count

27. A Voting Problem using structure and file.

fp = fopen("kibria.txt", "r");

fclose(fp);

fprintf(fp, "Golam Kibria %d", num);

int: ঠিক কতটি ক্যারেক্টার লিখা হলো সেই সংখ্যা রির্টান করে।

fscanf(fp, "%d", &num);

int: ঠিক কতটি ইনপুট পড়া হলো সেই সংখ্যা রির্টান করে।

sscanf(source string, "%d", &num);

int: ঠিক কতটি ইনপুট পড়া হলো সেই সংখ্যা রির্টান করে।

perror("File opening failed");

void: কোনো মান রির্টান করে না।

fgetc(fp);

int: একটি ক্যারেক্টারকে ইন্টিজার এ টাইপ কাষ্ট করে রির্টান করে।

fputc(ch, fp\_out);

int: যে ক্যারেক্টারটি লিখা হবে সেটিই রির্টান করে।

fgets(line, 80, fp\_in); //char line[80]

fgets(name);

fputs(name, fp\_in);

fseek(fp, long int offset, int whence)

long int offset : whence হতে কত বাইট পড়ে তা নির্দেশ করে।

int whence : যেই অবস্থানের সাথে offset যোগ করে হবে তার মান।

ftell(fp\_in);

return type long int. formet specifier %ld

মূলত ফাইল এর সাইজ নির্নয় করার জন্য ftell ব্যবহার হয়।

/\*Noel Kalicharan sir এর বইয়ের সমস্যা গুলোর শুধু সমাধান দেওয়া আছে। কোশ্চেন গুলো বই থেকে দেখে নিও। তাহলে প্রোগ্রাম গুলো বুঝতে সুবিধা হবে\*/

1. Inreoduction to file

Output:

compiler:

File is opened

file :

null(কিছুই না)

#include <stdio.h>

int main()

{

FILE\* file;

file = fopen("test.txt", "w");

if (file == NULL)

printf("File does not exist\n");

else

printf("File is opened\n");

fclose(file);

getch();

}

2. Writing a file using fputc()

#include <stdio.h>

int main()

{

Output:

compiler:

File is opened

File is written successfullly

file:

Golam Kibria

FILE\* file;

file = fopen("test.txt", "w");

char name[30] = "Golam Kibria";

int length = strlen(name);

int i;

if (file == NULL)

printf("File does not exist\n");

else

{

printf("File is opened\n");

for (i = 0; i < length; i++)

fputc(name[i], file);

printf("File is written successfullly\n");

}

fclose(file);

getch();

}

3. Writing a file using fputs()

#include <stdio.h>

int main()

{

FILE\* file;

char name[30];

file = fopen("test.txt", "w");

if (file == NULL)

{

printf("File does not exist\n");

}

else

{

printf("File is opened\n");

printf("Enter your name : ");

gets(name);

**/\*ay 2 line upore char er nichy dileo hoto\*/**

fputs(name, file);

printf("File is written successfully\n");

}

fclose(file);

}

output:

compiler:

File is opened

Enter your name : Golam kibria

File is written successfully

File

Golam Kibria

4. Writing to file using fprintf()

#include <stdio.h>

int main()

{

FILE\* file;

char name[30];

int age;

file = fopen("test.txt", "a");

if (file == NULL)

printf("File does not exist\n");

else

{

printf("File is opened\n");

printf("Enter your name : ");

gets(name);

printf("Enter your age : ");

scanf("%d", &age);

fprintf(file, "Name = %s, Age = %d\n", name, age);

printf("File is written successfully\n");

}

fclose(file);

}

Output:

Compiler:

File is opened

Enter your name : Golam Kibria

Enter your age : 20

File is written successfully

File :

Name = Golam Kibria, Age = 20

5. Reading a file using fgetc()

#include <stdio.h>

int main()

{

FILE\* file;

char ch;

file = fopen("test.txt", "r");

if (file == NULL)

{

printf("File does not exist\n");

}

else

{

printf("File is opened\n");

while (!feof(file))

{

ch = fgetc(file);

printf("%c", ch);

}

}

fclose(file);

}

**/\*protom a file a input dity hoby then oita compilar a show korby\*/**

Output:

FIle:

Name = kibria, Age = 20

Compiler :

File is opened

Name = kibria, Age = 20

6. Reading a file using fgets()

#include <stdio.h>

int main()

{

FILE\* file;

char ch[40];

file = fopen("test.txt", "r");

if (file == NULL)

{

printf("File does not exist\n");

}

else

{

printf("File is opened\n");

while (!feof(file))

{

fgets(ch, 39, file);

printf("%s\n", ch);

}

}

fclose(file);

}

Output:

File:

Name = kibria, Age = 20

Compiler :

File is opened

Name = kibria, Age = 20

7. Reading a file using fscanf()

#include<stdio.h>

int main()

{

FILE\* file;

char fname[40];

char lname[40];

int age;

file = fopen("test.txt", "r");

if (file == NULL)

{

printf("File does not exist");

}

else

{

printf("File is opened\n");

fscanf(file, "%s\t%s\t%d", &fname, &lname, &age);

printf("%s\t%s\t%d\n", fname, lname, age);

}

fclose(file);

}

Output:

File:

Golam Kiria 20

Compiler:

Golam Kibria 20

8. Students details by using file.

#include <stdio.h>

int main()

{

FILE\* file;

char name[30];

int age, phonenumber, num, i;

file = fopen("student.txt", "a");

if (file == NULL)

{

printf("File does not exist\n");

}

else

{

printf("File is opened\n");

printf("Enter numbers of students = ");

scanf("%d", &num);

for (i = 1; i <= num; i++)

{

printf("Enter student name = ");

scanf("%s", &name);

printf("Enter student age = ");

scanf("%d", &age);

printf("Enter students phone number = ");

scanf("%d", &phonenumber);

fprintf(file, "\n%s\t\t%d\t%d\n", name, age, phonenumber);

}

fclose(file);

}

Output:

Compilar:

File is opened

Enter numbers of students = 2

Enter student name = kibria

Enter student age = 20

Enter students phone number = 9878

Enter student name = saim

Enter student age = 20

Enter students phone number = 7654

File:

Name Age Phonenumber

kibria 20 9878

saim 20 7654

getch();

}

9. Preprocessor #define

#include <stdio.h>

#define name "Golam Kibria"

int main()

{

printf("%s\n", name);

}

Output: Golam Kibria

Or,

#include <stdio.h>

#define du printf("Daffodil IU")

int main()

{

du;

}

Output: Daffodil IU

Or,

#include <stdio.h>

#define du printf("Daffodil IU\n")

int main()

{

du;

du;

du;

}

Output:

Daffodil IU

Daffodil IU

Daffodil IU

10. Preprocessor creating header file

#include<stdio.h>

#include "info.h"

int main ()

{

printf ("Name = %s\n", name);

printf ("Age = %d\n", age);

getch();

}

#define name "Golam Kibria"

#define age 20

info.h namok header file er moddy ay information gulo asy.

11(1).writing to a file using fprintf()

#include <stdio.h>

Output:

File:

Golam Kibria

Dhaka, Bangladesh

int main()

{

FILE\* fp;

fp = fopen("myfile.txt", "w");

fprintf(fp, "Golam Kibria\n");

fprintf(fp, "Dhaka, Bangladesh\n");

fclose(fp);

}

11(2). Writing to a file using fprintf()

#include <stdio.h>

Output:

Golam Kibria

Dhaka, Bangladesh

Ki obsta

int main()

{

FILE\* fp;

fp = fopen("myfile.txt", "w");

fprintf(fp, "Golam Kibria\n");

fprintf(fp, "Dhaka, Bangladesh\n");

fclose(fp);

fp = fopen("myfile.txt", "a");

fprintf(fp, "Ki obsta\n");

fclose(fp);

}

12. Reading from a file using fscanf()

**/\*file input -> rur code -> file output\*/**

#include <stdio.h>

int main()

{

FILE\* in, \* out;

in = fopen("input.txt", "r");

out = fopen("output.txt", "w");

int num1, num2, sum;

fscanf(in, "%d", &num1);

fscanf(in, "%d", &num2);

sum = num1 + num2;

fprintf(out, "%d\n", sum);

fclose(in);

fclose(out);

}

Output:

* **প্রথমে Notepad দিয়ে একটা input.txt নামক টেক্সট ফাইল তৈরি করে নিতে হবে।**
* **তারপর input.txt ফাইল এ দুটি value(e.g.10 20) দিয়ে ফাইল টা সেভ করে নিতে হবে।**
* **সর্বশেষ, কোডটা রান করলে output.txt নামক একটা ফাইল অটোমেটিক তৈরি হবে এবং তার মধ্যে যোগফল(30) টা দেখা যাবে।**

13. নিজে নিজে হেডার ফাইল তৈরি করা

#include <stdio.h>

#include "dimik.h"

int main()

{

int n1 = 10, n2 = 5;

printf("%d + %d = %d\n", n1, n2, add(n1, n2));

printf("%d \* %d = %d\n", n1, n2, mul(n1, n2));

return 0;

}

Output:

10 + 5 = 15

10 \* 5 = 50

**/\*dimik.h নামক হেডার ফাইল এর মধ্যে নিচের তথ্য গুলো আছে\*/**

int add(int n1, int n2)

{

return n1 + n2;

}

int mul(int n1, int n2)

{

return n1 \* n2;

}

14. Sum of two number using file

**/\*file input -> rur code -> file output\*/**

#include <stdio.h>

int main()

{

FILE\* in, \* out;

in = fopen("inp.txt", "r");

out = fopen("out.txt", "w");

int n1, n2, sum;

char line[80];

fgets(line, 80, in);

printf("%s\n", line);

sscanf(line, "%d %d", &n1, &n2);

sum = n1 + n2;

fprintf(out, "%d\n", sum);

fclose(in);

fclose(out);

}

Output:

* **প্রথমে Notepad দিয়ে একটা inp.txt নামক টেক্সট ফাইল তৈরি করে নিতে হবে।**
* **তারপর input.txt ফাইল এ দুটি value(e.g.10 25) দিয়ে ফাইল টা সেভ করে নিতে হবে।**
* **সর্বশেষ, কোডটা রান করলে out.txt নামক একটা ফাইল অটোমেটিক তৈরি হবে এবং তার মধ্যে যোগফল(35) টা দেখা যাবে।**

15. Copy a image from input.img to output.img

#include <stdio.h>

#include <stdlib.h>

int main()

{

FILE\* in, \* out;

in = fopen("kibria.png", "rb");

out = fopen("image.png", "wb");

int x;

if (in == NULL)

{

perror("File opening failed");

return EXIT\_FAILURE:

}

while (1)

{

x = fgetc(in);

if (x == EOF) {

break;

}

fputc(x, out);

}

fclose(in);

fclose(out);

}

Output:

image.png নামক একটা নতুন ফাইল তৈরি হয়ে

kibria.png এর মধ্যে যে ছবিটা ছিলো তা দেখিয়েছে।

16. Fingign sum and aveage using file. N109

#include <stdio.h>

int main()

{

FILE\* in;

in = fopen("inp.txt", "r");

int num, sum = 0, n = 0;

fscanf(in, "%d", &num);

if (num == 0) printf("No numbers were entered\n");

else

{

while (num != 0)

{

n = n + 1;

sum = sum + num;

fscanf(in, "%d", &num);

}

printf("%d numbers were entered\n", n);

printf("The sum is = % d\n", sum);

printf("The average is = %.2lf\n", (double)sum / n);

}

fclose(in);

}

Output:

file: 24 13 55 32 10 0

compiler:

5 numbers were entered

The sum is = 134

The average is = 26.80

17. Finging sum and average using fileN112

**/\*file input -> run code -> file output\*/**

#include <stdio.h>

int main()

{

FILE\* in, \* out;

in = fopen("inp.txt", "r");

out = fopen("out.txt", "w");

int num, sum = 0, n = 0;

fscanf(in, "%d", &num);

if (num == 0) fprintf(out, "No numbers were entered\n");

else

{

while (num != 0)

{

n = n + 1;

sum = sum + num;

fscanf(in, "%d", &num);

}

fprintf(out, "%d numbers were entered\n", n);

fprintf(out, "The sum is = % d\n", sum);

fprintf(out, "The average is = %.2lf\n", (double)sum / n);

}

fclose(in);

fclose(out);

}

Output:

int.txt

24 13 55 32 10 0

out.txt

5 numbers were entered

The sum is = 134

The average is = 26.80

18. Usng of fseek and perror function.

#include <stdio.h>

#include <stdlib.h>

int main()

{

FILE\* fp;

fp = fopen("in.txt", "r");

int ch;

if (fp == NULL)

{

perror("Can not open file");

return EXIT\_FAILURE:

}

ch = fgetc(fp);

printf("%c\n", (char)ch);

ch = fgetc(fp);

printf("%c\n", (char)ch);

fseek(fp, 0, 0);

ch = fgetc(fp);

printf("%c\n", (char)ch);

fclose(fp);

}

File:

ABCKEFGHIJK

MNOPQRST

UNWX

YZ

**Output:**

compiler:

A

B

A

* fseek(fp, 0, SEEK\_SET) same output like A\nB\nA
* fseek(fp, 0, SEEK\_CUR) output: A\nB\nC
* fseek(fp, 0, SEEK\_END) output: A\nB
* fseek(fp, sizeof(char) \* 3, SEEK\_CUR); output: A\nB\nF/

তিনটি ক্যারেক্টার বাদ দিয়ে আউটপুট দেখাবে।

19. Finding a file size using ftell()

#include <stdio.h>

#include <stdlib.h>

int main()

{

FILE\* in;

in = fopen("kibria.png", "rb");

if (in == NULL)

{

perror("File opening failed");

return EXIT\_FAILURE;

}

fseek(in, 0, SEEK\_END);

printf("File size: %ld bytes\n", ftell(in));

printf("File size(kilobytes): %ld\n", ftell(in) / 1024);

fclose(in);

}

Output:

compiler:

**kibria.png নামক image টার সাইজ বের করে দিবে।**

20. Remove a file

#include <stdio.h>

int main()

{

int return\_value;

char\* filename = "kibria.png";

return\_value = remove(filename);

if (return\_value != 0)

{

perror("File remove Failed");

return 1;

}

printf("%s removed successfully\n", filename);

}

Output:

kibria.png নামক .png ফাইল টা পুরো সি ফোল্ডার হতে কোড টা রান করার সাথে সাথে বিমুভ হয়ে যাবে।

21. Payroll N117

#include <stdio.h>

#include <string.h>

#define maxreghour 40

#define overtimefactor 1.5

int main()

{

FILE\* inp = fopen("paydata.txt", "r");

FILE\* out = fopen("payroll.txt", "w");

char firstname[30], lastname[30], name[30], bestpaid[30];

double hours, rate, regularpay, overtimepay, netpay;

double totalbill = 0, mostpaid = 0;

int numofemploye = 0;

fprintf(out, "Name\t\t\t Hours\t Rate\t\tRegular\t\tOvertime\tNet\n\n");

//2 space & 3 space

fscanf(inp, "%s", firstname);

while (strcmp(firstname, "END") != 0)

{

numofemploye++;

fscanf(inp, "%s %lf %lf", lastname, &hours, &rate);

if (hours <= maxreghour)

{

regularpay = hours \* rate;

overtimepay = 0;

}

else

{

regularpay = maxreghour \* rate;

overtimepay = (hours - maxreghour) \* rate \* overtimefactor;

}

netpay = regularpay + overtimepay;

strcpy(name, firstname); strcat(name, " "); strcat(name, lastname);

fprintf(out, "%-15s %14.1lf %10.2lf", name, hours, rate);

fprintf(out, "%14.2lf %15.2lf %12.2lf\n", regularpay, overtimepay, netpay);

if (netpay > mostpaid)

{

mostpaid = netpay;

strcpy(bestpaid, name);

}

totalbill = totalbill + netpay;

fscanf(inp, "%s", firstname);

}

fprintf(out, "\nNumbers of employess: %d\n", numofemploye);

fprintf(out, "Total bill: %.2lf\n", totalbill);

fprintf(out, "%s earned the most pay of $%.2lf\n", bestpaid, mostpaid);

fclose(inp);

fclose(out);

}

**Output:**

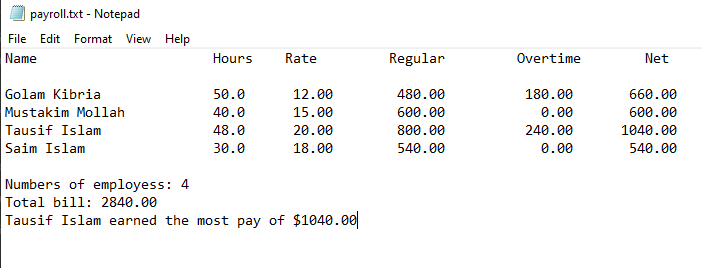
Golam Kibria 50.0 12.

Mustakim Mollah 40.0 15.00

Tausif Islam 48.0 20.00

Saim Islam 30.0 18.00

END



22. Read character from a file N156

#include <stdio.h>

int main()

{

FILE\* in;

in = fopen("input.txt", "r");

char ch;

ch = getc(in);

while (ch != '\n')

{

putchar(ch);

ch = getc(in);

}

putchar('\n');

fclose(in);

}

***/\*putchar is used for write a single character to the standard output***

***and also write the character in the next position int the output\*/***

#include <stdio.h>

int main()

{

FILE\* in;

in = fopen("input.txt", "r");

char ch;

ch = getc(in);

while (ch != '\n' && ch != EOF)

{

putchar(ch);

ch = getc(in);

}

//putchar('\n');

fclose(in);

}

***/\*putchar('\n) is not needed cuz getc(in) will***

***returned EOF when the end of file is reached\*/***

#include <stdio.h>

int main()

{

FILE\* in;

in = fopen("input.txt", "r");

char ch;

ch = getc(in);

while (ch != EOF)

Output:

File: golam Kibria

Compiler: golam kibria

{

putchar(ch);

ch = getc(in);

}

fclose(in);

}

***/\*This will also print same output\*/***

***/\*All the three problem is same but the solving system is different\*/***

23. Numbering the lines N159

***/\*This program prints the data from a file by numbering\*/***

#include <stdio.h>

int main()

{

FILE\* in;

in = fopen("input.txt", "r");

char ch;

int linenumber = 0;

int writtenlinenumber = 1;

ch = getc(in);

while (ch != EOF)

{

if (writtenlinenumber == 1)

{

printf("%2d. ", ++linenumber);

writtenlinenumber = 0;

}

putchar(ch);

if (ch == '\n')

{

writtenlinenumber = 1;

}

ch = getc(in);

}

fclose(in);

}

***/\*The expression ++linenumber means linenumber is increment***

***friest brfore print(1, 2, 3). But if we use linenumber++ it***

***will print frist then increment(0, 1, 2)\*/***

Output:

File:

There was a little girl

Who had a little curl

Right in the middle of her head

Compiler:

1. There was a little girl

2. Who had a little curl

3. Right in the middle of her head

24. A voting problem N235

#include <stdio.h>

#include <string.h>

FILE\* in, \* out;

int main()

{

void initialize(char name[][31], int vote[]);

void printresult(char name[][31], int vote[], int validvote, int spoilvote);

char name[8][31];

int vote[8];

int validvote = 0, spoilvote = 0;

int x;

in = fopen("votee.txt", "r");

out = fopen("resultt.txt", "w");

initialize(name, vote);

fscanf(in, "%d", &x);

while (x != 0)

{

if (x < 1 || x > 7) {

fprintf(out, "Invalid Vote : %d\n", x);

++spoilvote;

}

else {

++vote[x];

++validvote;

}

fscanf(in, "%d", &x);

}

printresult(name, vote, validvote, spoilvote);

fclose(in);

fclose(out);

}

void initialize(char name[][31], int vote[])

{

char lastname[30];

for (int t = 1; t <= 7; t++)

{

fscanf(in, "%s %s", name[t], lastname);

strcat(name[t], " ");

strcat(name[t], lastname);

vote[t] = 0;

}

}

int getlargest(int num[], int m, int n)

{

int big = m;

for (int i = m + 1; i <= n; i++) {

if (num[i] > num[big])

big = i;

}

return big;

}

void printresult(char name[][31], int vote[], int validvote, int spoilvote)

{

int getlargest(int num[], int m, int n);

fprintf(out, "\nNumber of voters : %d\n", validvote + spoilvote);

fprintf(out, "Number of valid vote : %d\n", validvote);

fprintf(out, "Number of spoil vote : %d\n", spoilvote);

fprintf(out, "\nCandidate Score\n\n");

for (int p = 1; p <= 7; p++) {

fprintf(out, "%-19s %3d\n", name[p], vote[p]);

}

fprintf(out, "\nThe winner\n");

int win = getlargest(vote, 1, 7);

int winningvote = vote[win];

for (int n = 1; n <= 7; n++) {

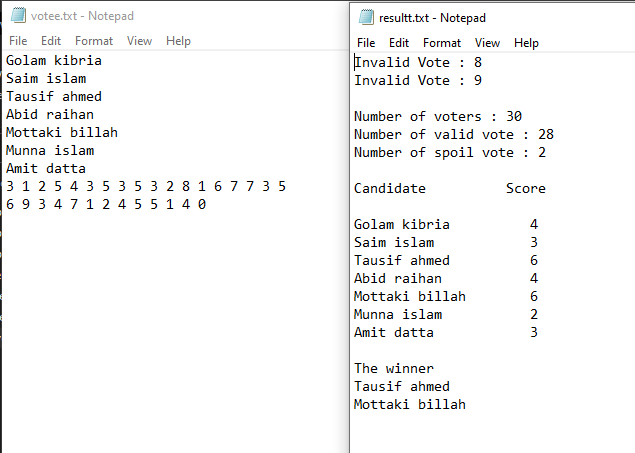
if (vote[n] == winningvote) {

fprintf(out, "%s\n", name[n]);

}

}

}



25. A Geography Quiz Problem N230

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int main()

{

void getString(FILE\*, char[]);

void askonequestion(char[], char[], char[]);

char country[51];

char capital[51], CAPITAL[51];

char endofdata[] = "\*";

FILE\* in = fopen("quizdata.txt", "r");

if (in == NULL) {

printf("Cannot find file\n");

exit(1); //exit is used to terminate execution of a program.

}

getString(in, country);

while (strcmp(country, endofdata) != 0) {

getString(in, capital);

getString(in, CAPITAL);

askonequestion(country, capital, CAPITAL);

getString(in, country);

}

}

void askonequestion(char country[], char capital[], char CAPITAL[])

{

void lettersonlyupper(char[], char[]);

char answer[51], ANSWER[51];

printf("\nWhat is the capital of %s? ", country);

gets(answer);

lettersonlyupper(answer, ANSWER);

if (strcmp(ANSWER, CAPITAL) == 0)

printf("Correct\n");

else {

printf("Wrong. Try again\n");

printf("\nWhat is the capital of %s? ", country);

gets(answer);

lettersonlyupper(answer, ANSWER);

if (strcmp(ANSWER, CAPITAL) == 0)

printf("Correct\n");

else

printf("Wrong. Answer is : %s\n", capital);

}

}

void lettersonlyupper(char word[], char WORD[])

{

int i = 0, n = 0;

char ch;

while ((ch = word[i++]) != '\0') {

if (isalpha(ch))

WORD[n++] = toupper(ch);

}

WORD[n] = '\0';

}

void getString(FILE\* in, char str[])

{

char ch, delimiter;

int x = 0;

str[0] = '\0';

while (isspace(ch = getc(in))); //empty while body.

if (ch == EOF)

return;

delimiter = ch;

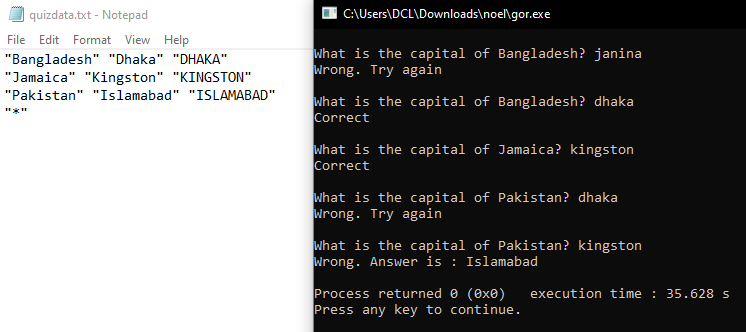
while (((ch = getc(in)) != delimiter) && (ch != EOF)) {

str[x++] = ch;

}

str[x] = '\0';

}



26. Word frequency count N-267

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int main()

{

int getWord(FILE \* in, char str[]);

int binarysearch(int first, int last, char key[], int max, char list[][max]);

void addtolist(char item[], int max, char list[][max], int freq[], int p, int n);

void printresult(FILE \* out, int max, char list[][max], int freq[], int n);

char wordlist[50][11];

char perword[11];

int frequency[50];

int count = 0;

FILE\* in = fopen("passage.txt", "r");

if (in == NULL) {

printf("Cannot find file\n");

exit(1);

}

FILE\* out = fopen("output.txt", "w");

if (out == NULL) {

printf("Cannot create output file\n");

exit(2);

}

for (int h = 1; h <= 50; h++)

frequency[h] = 0;

while (getWord(in, perword) != 0) {

int location = binarysearch(0, count - 1, perword, 11, wordlist);

if (strcmp(perword, wordlist[location]) == 0)

++frequency[location]; /\*Word found\*/

else { /\*This is a new word\*/

if (count < 50) {

addtolist(perword, 11, wordlist, frequency, location, count - 1);

++count;

}

else

fprintf(out, "'%s' not added to table\n", perword);

}

}

printresult(out, 11, wordlist, frequency, count);

}

int getWord(FILE\* in, char str[])

{

***/\*Store the next word, if any, in str. Convert word to lowercase***

***Retrun 1 if a word is found, if not found then return 0\*/***

char ch;

int n = 0;

/\*Read until white space\*/

while (!isalpha(ch = getc(in)) && ch != EOF); ***/\*Empty while body\*/***

if (ch == EOF) return 0;

str[n++] = tolower(ch);

while (isalpha(ch = getc(in)) && ch != EOF) {

if (n < 10)

str[n++] = tolower(ch);

}

str[n] = '\0';

return 1;

}

int binarysearch(int first, int last, char key[], int max, char list[][max])

{

***/\*Search for key from list[first] to list[last] . If key is found returns keys location. If not found, return the location in which is should be inserted. The callind program will check the location to determine if found\*/***

while (first <= last) {

int mid = (first + last) / 2;

int cmp = strcmp(key, list[mid]);

if (cmp == 0)return mid;

if (cmp < 0)last = mid - 1;

else first = mid + 1;

}

return first; ***/\*Not found; should be inserted in location first\*/***

}

void addtolist(char item[], int max, char list[][max], int freq[], int p, int n)

{

***/\*Adds item in position list[p]; sets freq[p] to 1.***

***Shift list[n] down to list [p] to the right\*/***

for (int h = n; h >= p; h--) {

strcpy(list[h + 1], list[h]);

freq[h + 1] = freq[h];

}

strcpy(list[p], item);

freq[p] = 1;

}

void printresult(FILE\* out, int max, char list[][max], int freq[], int n)

{

fprintf(out, "Words\t\tFrequency\n\n");

for (int h = 0; h < n; h++)

fprintf(out, "%-13s %4d\n", list[h], freq[h]);

}



27. A Voting Problem using structure and file. N301

#include <stdio.h>

#include <string.h>

typedef struct {

char name[31];

int numofvotes;

}persondata;

persondata candidate[7];

typedef struct {

int valid, spoilt;

}votecount;

int main()

{

void initialize(persondata person[], int max, FILE \* in);

votecount processvotes(persondata person[], int max, FILE \* in, FILE \* out);

void printresult(persondata person[], int max, votecount c, FILE \* out);

persondata candidate[8];

votecount count;

FILE \*in = fopen("votes.txt", "r");

FILE\* out = fopen("result.txt", "w");

initialize(candidate, 7, in);

count = processvotes(candidate, 7, in, out);

printresult(candidate, 7, count, out);

fclose(in);

fclose(out);

} ***/\*End main\*/***

void initialize(persondata person[], int max, FILE\* in)

{

char lastname[31];

for (int h = 1; h <= max; h++) {

fscanf(in, "%s %s", person[h].name, lastname);

strcat(person[h].name, " ");

strcat(person[h].name, lastname);

person[h].numofvotes = 0;

}

}  ***/\*End initialize\*/***

votecount processvotes(persondata person[], int max, FILE\* in, FILE\* out)

{

votecount temp;

temp.valid = temp.spoilt = 0;

int v;

fscanf(in, "%d", &v);

while (v != 0) {

if (v<1 || v>max) {

fprintf(out, "Invalid vote: %d\n", v);

++temp.spoilt;

}

else {

++person[v].numofvotes;

++temp.valid;

}

fscanf(in, "%d", &v);

}

return temp;

} ***/\*End processvotes\*/***

int getlargest(persondata person[], int first, int last)

{

***/\*Return the index of highest votes from person[first] to person[last]\*/***

int big = first;

for (int h = first + 1; h <= last; h++) {

if (person[h].numofvotes > person[first].numofvotes)

big = h;

}

return big;

} ***/\*End getlargest\*/***

void printresult(persondata person[], int max, votecount c, FILE\* out)

{

int getlargest(persondata person[], int first, int last);

fprintf(out, "\nNumber of votes: %d\n", c.valid+c.spoilt);

fprintf(out, "Number of valid votes: %d\n", c.valid);

fprintf(out, "Number of spoilt votes: %d\n", c.spoilt);

fprintf(out, "\nCandidate score\n\n");

for (int h = 1; h <= max; h++)

fprintf(out, "%-15s %3d\n", person[h].name, person[h].numofvotes);

fprintf(out, "\nThe winner\n");

int win = getlargest(person, 1, max);

int winningvote = person[win].numofvotes;

for (int h = 1; h <= max; h++) {

if (person[h].numofvotes == winningvote)

fprintf(out, "%s\n", person[h].name);

}

} ***/\*End printresult\*/***

